Master's Thesis

Title

Studies on Modeling Packet Delay Dynamics of the Internet using System Identification and its Application for Designing a Rate-Based Congestion Control Mechanism

Supervisor

Prof. Masayuki Murata

Author

Mitsushige Morita

February 12 2002

Department of Informatics and Mathematical Science,

Graduate School of Engineering Science,

Osaka University

Master's Thesis

Studies on Modeling Packet Delay Dynamics of the Internet using System Identification and its Application for Designing a Rate-Based Congestion Control Mechanism

Mitsushige Morita

Abstract

Understanding the end-to-end packet delay dynamics of the Internet is of crucial importance since it directly affects the QoS (Quality of Services) of realtime services, and it enables us to design an efficient congestion control mechanism. In this thesis, we measure the round-trip time in the actual operating Internet, and build a mathematical model representing its dynamics using system identification. First, we regard the network, seen by a specific source host, as a dynamic SISO (Single-Input and Single Output) system. The input to the system is an *instantaneous* packet transmission rate from the source host during a fixed sampling interval. Also the output is an *instantaneous* average round-trip time observed by the source host during a fixed sampling interval. ICMP (Internet Control Message Protocol) is utilized to measure the round-trip time for each packet. We model the round-trip time dynamics using the ARX (Auto-Regressive eXogenous) model. The coefficients of the ARX model are determined from the input and output data collected in LAN and WAN environments using system identification. Through numerical examples, We show that in LAN environment, the round-trip time dynamics can be accurately modeled by the ARX model. We also show that in WAN environment, the round-trip time dynamics can be accurately modeled when the bottleneck link is shared by a small number of users. Our findings indicate that the round-trip time dynamics can be accurately modeled in such a network where the packet transmission rate from a source host has strong correlation with

the round-trip time. It is indispensable to control the packet transmission rate from a source host for preventing packet losses using the obtained model. We therefore design a simple rate control mechanism by applying the state feedback used in the classical control theory to the model of the round-trip time dynamics. By dynamically controlling the packet transmission rate from the source host for stabilizing the round-trip time, it is possible to achieve an efficient end-to-end transmission as well as to avoid packet losses in the network. Through simulation experiments, I quantitatively evaluate the efficiency of our rate control mechanism.

Keywords

Delay-based Congestion Control, Control Theory, System Identification, Packet Delay

Contents

List of Figures

1 Introduction

Understanding the end-to-end packet delay dynamics of the Internet is of crucial importance since (1) it directly affects the QoS (Quality of Services) of realtime applications, and (2) it enables us to design an efficient congestion control mechanism for both realtime and non-realtime applications. For non-realtime applications, a delay-based approach for congestion control mechanisms, rather than a loss-based approach as used in TCP (Transmission Control Protocol), has been proposed (e.g., [1, 2]). The main advantage of such a delay-based approach is, if it is properly designed, packet losses can be prevented by anticipating impending congestion from increasing packet delays.

For a long time, queueing theory has been extensively used as a powerful tool to analyze packet-switched networks. In general, the queueing theory assumes stationarity of the network, and allows us to obtain several performance measures such as the average packet delay and the average packet loss probability. However, the stringent limitation of the queuing theory is its difficulty to analyze the *dynamic behavior* of the network. Several measurement-based studies suggest that the end-to-end packet behavior in the Internet is quite dynamic [3-5]. Another approach, being different from the queueing theory, should therefore be taken to investigate the packet delay dynamics of the Internet.

In this thesis, we propose a novel approach to model the end-to-end packet delay dynamics of the Internet. The main idea of our approach is treating the network, seen by a specific source host, as a *black-box*, and modeling the end-to-end packet delay dynamics using *system identification* [6]. The end-to-end packet delay dynamics are modeled as a SISO (Single-Input and Single-Output) system based on the ARX (Auto-Regressive eXogenous) model. The input to the system is an *instantaneous* packet transmission rate from the source host during a fixed sampling interval. Also the output is an *instantaneous* average round-trip time observed by the source host during a fixed sampling interval. We collect the input and output data from real networks, and model the round-trip time dynamics of the Internet by system identification.

After discussing advantages and disadvantages of several measurement methods for the round-trip time, We present a measurement method using ICMP (Internet Control Message Protocol) to collect the input and output data for determining the coefficients of the ARX model. Since almost all hosts and routers respond to ICMP packets, this method can be used in various network environments. In this thesis, we collect input and output data for system identification in LAN and WAN environments, and build a model for the round-trip time dynamics. Using the obtained data, coefficients of the ARX model are determined using the least-square method. we then investigate how accurately the ARX model can represent the round-trip time dynamics of the Internet. We evaluate the accuracy of the ARX model in time domain; that is, we compare the simulated outputs from the ARX model (i.e., round-trip times) with the actual round-trip times. We also examine the model accuracy in frequency domain using a spectral analysis. Through numerical examples, we show that in LAN environment, the round-trip time dynamics can be accurately modeled by the ARX model. We also show that in WAN environment, the round-trip time dynamics can be accurately modeled when the bottleneck link is shared by a small number of users. These results indicate that the round-trip time can be modeled in the network, where the transmission rate has great effects on the round-trip time. Therefore, using the obtained model, it is possible to control the transmission rate to prevent packet loss. We design a simple rate control mechanism by applying the state feedback used in the classical control theory to the model of the round-trip time dynamics. By dynamically controlling the packet transmission rate from the source host for stabilizing the round-trip time, it is possible to achieve an efficient end-to-end transmission as well as to avoid packet losses in the network. Through simulation experiments, I quantitatively evaluate the efficiency of our rate control mechanism.

This thesis is organized as follows. In Section 2, we summarize related works in recent publications. In Section 3, a black-box approach for modeling the round-trip time dynamics of the Internet using the ARX model is explained. In Section 4, we discuss several measurement methods of the round-trip time, in particular, for collecting the input and output data for system identification. Section 5 shows several measurement and modeling results, and discuss how accurately the ARX model can capture the round-trip time dynamics. In Section 6, we discuss several possible applications of our approach. In Section 7, we conclude this thesis.

2 Related Works

In the literature, there have been several measurement-based studies regarding the end-to-end packet delay [3, 4, 7, 8] and the end-to-end path characteristics [5, 9]. In [3], the authors have examined the end-to-end packet delay and loss behavior in the Internet using small UDP probe packets. In [4], the authors have examined the correlation between packet delay and packet loss experienced by a continuous-media traffic source, based on measurements of per-packet delays and packet loss. In [7], a large sets of TCP measurements have been used to discuss two estimation problems: estimation of the retransmission timer (RTO) for a TCP connection, and estimation of the available bandwidth. In [8], the authors have presented an approach to characterize loss and delay behavior on a transmission link based on end-to-end multicast measurements. In [5], the packet dynamics of the Internet have been analyzed based on measurements of about 20,000 TCP data transfers. In [9], the routing behavior of the Internet has been analyzed based on measurements of about 40,000 end-to-end *traceroute* results. However, those studies are limited to analyse a statistical behavior of the end-to-end packet delays and/or path characteristics. In other words, the dynamics of the packet delay of the Internet, which is the main concern of this thesis, has not been investigated.

Aside from analyses of the end-to-end packet delay, another area of measurement-based studies is regarding a black-box modeling of the network traffic [10-14]. In [10], the authors have proposed a traffic model for wide-area TCP traffic by characterizing several distributions of, for example, the packet inter-arrival time and the number of bytes transferred. In [11], the authors have proposed a fast algorithm to construct a CMRP (Circulant Modulated Rate Process) for traffic modeling. In [12], CMRP and ARMA (Auto-Regressive

Moving Average) have been discussed as a traffic model. In [13, 14], a measurement-based tool for traffic modeling and queueing analysis has been developed, which uses CMPP (Circulant Modulated Poisson Process) for a traffic model. Those studies are closely related to our black-box modeling approach treadted in this thesis, but there is a significant difference. Those studies have focused on traffic modeling based only on outputs (i.e., observed amount of traffic). On the contrary, this thesis focuses on modeling the round-trip time dynamics based on both inputs (i.e., packet inter-departure time) and outputs (i.e., round-trip time variation). In other words, we focus on how the round-trip time of a packet sent from a source host is affected by its past packet transmission process.

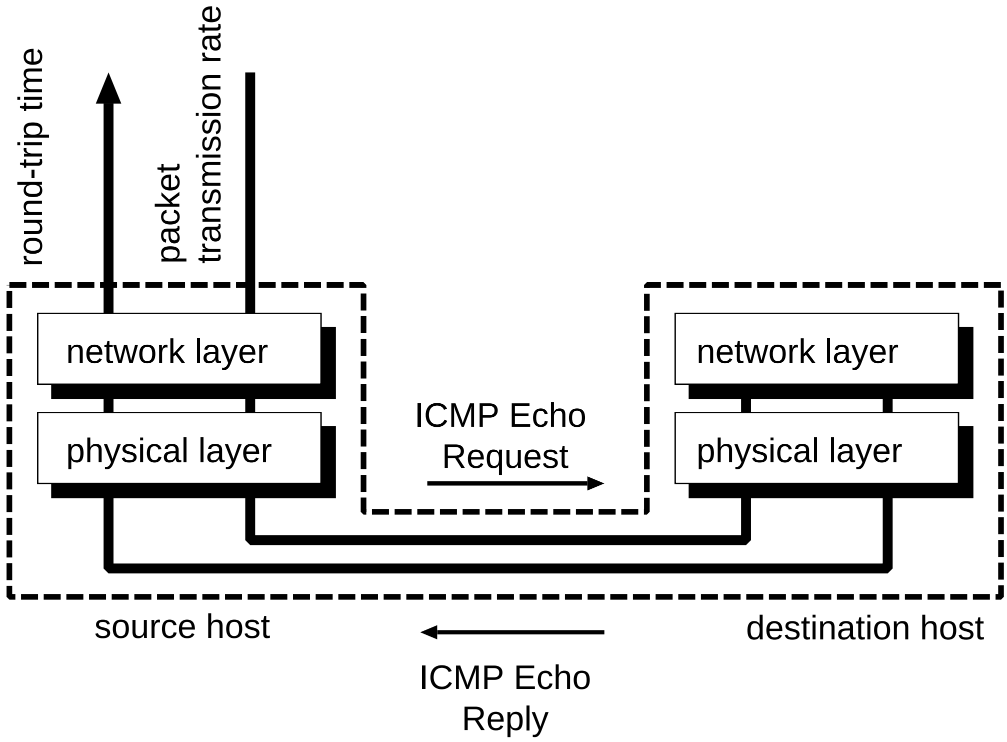


Figure 1: Modeling round-trip time dynamics as SISO system.

3 Black-Box Modeling using ARX Model

3.1 Black-Box Modeling

As depicted in Fig. 1, the network seen by a specific source host, including underlying protocol layers (e..g, physical, data-link, and network layers), is considered as a black-box. Our goal of this thesis is to build a SISO system describing the round-trip time dynamic which is represented by the relation between a packet sending process from the source host and its resulting round-trip time observed at the source host.

In this thesis, the ARX model is used and its coefficients are determined using system identification [6]. Figure 2 illustrates a fundamental concept of using the ARX model for

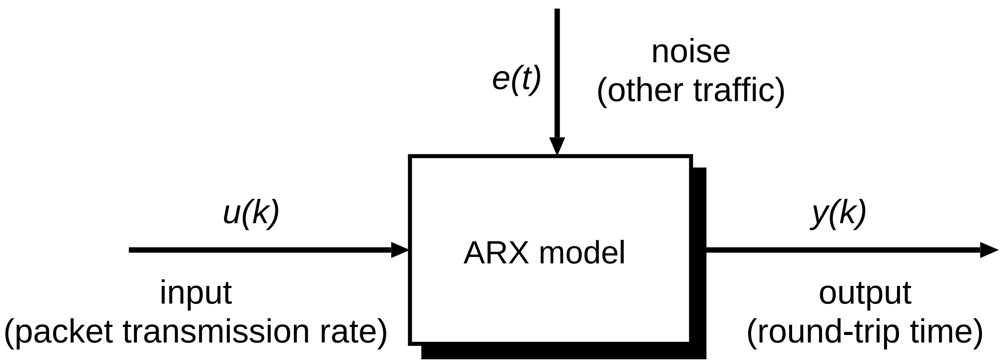


Figure 2: ARX model for modeling round-trip time dynamics.

modeling the packet delay dynamics. The input to the ARX model is a packet transmission rate from the source host, and the output from the ARX model is a round-trip time measured by the source host. Effects of other traffic (i.e., packets coming from other hosts) are modeled as the noise to the ARX model. Letting $u(k)$ and $y(k)$ be the input and the output data at slot k, the ARX model is defined as

$$A(q)\, y(k) \quad = \quad B(q)\, u(k - n_d) + e(k) \tag{1}$$

where $A(q)$ and $B(q)$ are given by

$$\begin{aligned} A(q) &= 1 + a_1 q^{-1} + \ldots + a_{n_a} q^{-n_a} \\ B(q) &= b_1 + b_2 q^{-1} + \ldots + b_{n_b} q^{-n_b+1} \end{aligned}$$

In the above equations, $e(k)$ is unmeasurable disturbance (i.e., noise), and q^{-1} is the delay operator; i.e., $q^{-1}u(k) \equiv u(k-1)$. The numbers n_a and n_b are the orders of polynomials. The number n_d corresponds to delays from the input to the output. For compact description,

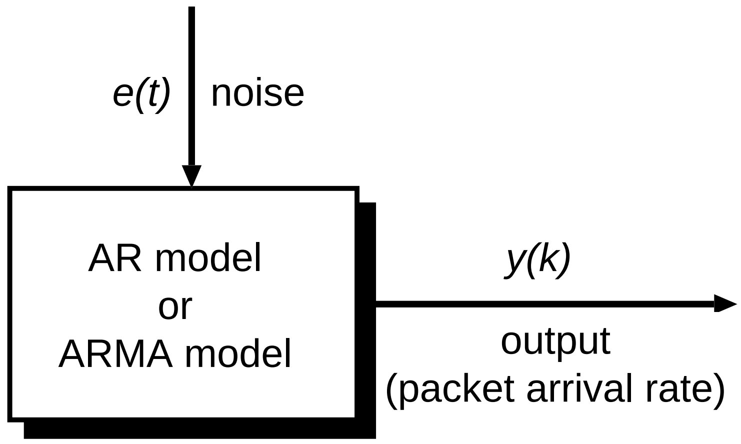


Figure 3: AR model or ARMA model for modeling network traffic.

ζ and θ are introduced as

$$\begin{aligned} \zeta &= [n_a, n_b, n_d] \\ \theta &= [a_1, \ldots, a_{n_a}, b_1, \ldots b_{n_b}]^T \end{aligned}$$

Our approach of a black-box modeling using the ARX model is distinctive from other black-box approaches, which model network traffic using the AR (Auto-Regressive) model or the ARMA (Auto-Regressive Moving Average) model [12, 15, 16]. Figure 3 illustrates a typical usage of the AR model or the ARMA model for modeling network traffic. Comparing Figs. 2 and 3, the ARX model has the input whereas either the AR model or the ARMA model does not. In other words, only the ARX model can represent the dynamics, i.e., how the past input data affects the future output data.

Note that the ARX model has a drawback in modeling the round-trip time dynamics; i.e., the ARX model is a linear time-invariant model, so it cannot rigorously capture non-linearity of the round-trip time dynamics. But it should be noted that the ARX model is applicable in various control engineering problems. This is because non-linear dynamical systems operating around the stable point can be well approximated by a linear system [6]. In Section 5, we will investigate how accurately the round-trip time dynamics can be described by the ARX model.

3.2 Input and Output Selection

It is essentially to define the input and output carefully, since a SISO system is built only from the measured input and output data. For system identification purposes, it is desirable [6]:

1. the input depends on the output
2. the target system is not a feed-back system
3. the noise to the system equals zero on average, and is stable

For our purposes, the above conditions can be interpreted as

1. the input to the system affects the round-trip time
2. the input to the system is independent of past round-trip times
3. the noise occured in the network equals to zero on average, and is stable

Since the round-trip time is dependent on the congestion status of the network [4], according to first condition, we choose the packet transmission rate as the input to the system. In general, as the packet transmission rate from the source host increases, the round-trip time increases. On the contrary, as the packet transmission rate decreases, the round-trip time decreases. Since most of system identification techniques assume independence between the input and output data, the second condition is necessary for modeling the round-trip time dynamics. Although it is possible to apply system identification to feedback systems, there are several limitations to apply to real network system. Third condition is related to how output from the system should be decided. For example, when we choose the round-trip time as the output from the system, the round-trip time is affected by the other traffic in the network.

In this thesis, we therefore define that the input to the system and the output from the system as the packet transmission rate and the average round-trip time during a fixed sampling interval, respectively. In general, a fixed sampling interval is assumed in system identification. In this thesis, we therefore use a fixed sampling interval for improving the model accuracy. More specifically, the input $u(k)$ and the output $y(k)$ are defined as follows. Let $t_s(i)$ be the time at which the ith packet is injected into the network, and $t_r(i)$ be the time at which the ith ACK packet is received by the source host. We further introduce $l(i)$ as the size of the ith packet including the IP header, and T as the sampling interval. Then, $u(k)$ and $y(k)$ are defined as

$$
\begin{aligned}
u(k) &= \frac{\sum_{i \in \phi_s(k)} l(i)}{T} \\
y(k) &= \frac{\sum_{i \in \phi_r(k)} (t_r(i) - t_s(i))}{|\phi_r(k)|}
\end{aligned}
$$

where $\phi_s(k)$ (or $\phi_r(k)$) is the set of packet numbers sent (or received) during kth sampling interval; i.e.,

$$
\phi_s(k) \equiv \{n : k\,T \leq t_s(n) < (k+1)\,T\}
$$

$$\phi_r(k) \equiv \{n : k\,T \leq t_r(n) < (k+1)\,T\}$$

3.3 System Identification

The system identification for the ARX model is formulated as a minimization problem, where the cost function is given by a loss function [6]. Because of space limitation, only the outline is shown in this thesis, and interested readers should refer to [6] for more detail.

Let $\psi(k)$ be a vector of all past n_a outputs and n_b inputs.

$$\begin{aligned}\psi(k) &= [-y(k-1), \ldots, -y(k-n_a), \\ &\quad u(k-n_d-1), \ldots, u(k-n_d-n_b)]^T \end{aligned} \tag{2}$$

Using Eq. (1), the output from the ARX model $\hat{y}(k|\theta)$ is given by

$$\hat{y}(k|\theta) = \psi^T(k)\,\theta \tag{3}$$

The loss function $V_N(\theta, Z^N)$ is defined as the sum of all squared prediction errors for N input and output data.

$$V_N(\theta, Z^N) = \frac{1}{N}\sum_{k=1}^{N}(y(k) - \hat{y}(k|\theta))^2 \tag{4}$$

where Z^n is the past input and output data defined as

$$Z^N = \{u(1), y(1), \ldots, u(N), y(N)\} \tag{5}$$

The solution $\hat{\theta}_N$ that minimizes the above loss function is easily obtained by the least squares method:

$$\hat{\theta}_N = \left[\sum_{k=1}^{N} \psi(k)\psi^T(k)\right]^{-1} \sum_{k=1}^{N} \psi(k)y(k) \tag{6}$$

4 Data Collection using ICMP Packet

4.1 Measurement Method

For collecting input and output data from a real network, it is necessary to send a series of probe packets into the network, and to measure their resulting round-trip times. For sending a probe packet, the following protocols can be used.

- TCP (Transmission Control Protocol)
- UDP (User Datagram Protocol)
- ICMP (Internet Control Message Protocol)

In what follows, we briefly discuss advantages and disadvantages of these protocols for sending a probe packet to collect input and output data, in particular, for system identification.

TCP has a feedback-based congestion control mechanism, which controls the packet sending process from a source host according to the congestion status of the network. Since it is an ACK-based protocol, it is easy for the source host to measure the round-trip time for each packet. However, because of such a feedback-based mechanism, TCP is not suitable for sending a probe packet for two reasons. First, although the input (i.e., the packet transmission rate) should contain diverse frequencies for system identification purposes, the packet transmission rate of TCP would have limited frequencies. Second, as described in Section 3, regardless of many system identification techniques assuming an independence between the input and the output, the independence assumption cannot be satisfied with TCP since the packet transmission rate is dependent on the past round-trip times.

On the contrary, UDP has no feedback-based control. The packet transmission rate of UDP can be freely controlled. However, UDP is a one-way protocol. The destination host must perform some procedure to measure the round-trip time for each packet at the source host. One possible way is to use *ICMP Destination Unreachable* message as in the *traceroute* program [17]. When the host receives a UDP packet to an unreachable port, it returns ICMP Destination Unreachable message to the source host. The source host can therefore measure the round-trip time by observing the elapsed time between the UDP packet transmission and the receipt of the corresponding ICMP packet. However, as specified in [18], generation of ICMP Destination Unreachable messages is limited to a low rate. Use of ICMP Destination Unreachable message is therefore not desirable to collect the input and output data for system identification.

ICMP is a protocol to exchange control messages such as routing information and node failures [19]. Since ICMP has no feedback-based control, the inter-departure time of ICMP packets can be freely controlled. Also it is easy to measure the round-trip time at the source host by using *ICMP Echo Request* and *ICMP Echo Reply* messages, as in the *ping* program. Although some network devices limit the rate of ICMP packets because of malicious use of them [20], such as a DoS (Denial of Service) attack, many network devices respond to ICMP Echo Request message and do not limit the rate.

In this thesis, we therefore choose the ICMP Echo message as a probe packet. More specifically, the source host sends a series of ICMP Echo Request messages to the destination host, and the destination host returns ICMP Echo Reply messages. We have modified the ping program to dynamically change the packet inter-departure time (originally fixed at one second). The destination host copies the payload of the received ICMP Echo Request message to the returning ICMP Echo Reply message. Thus, the ICMP Echo Reply packet

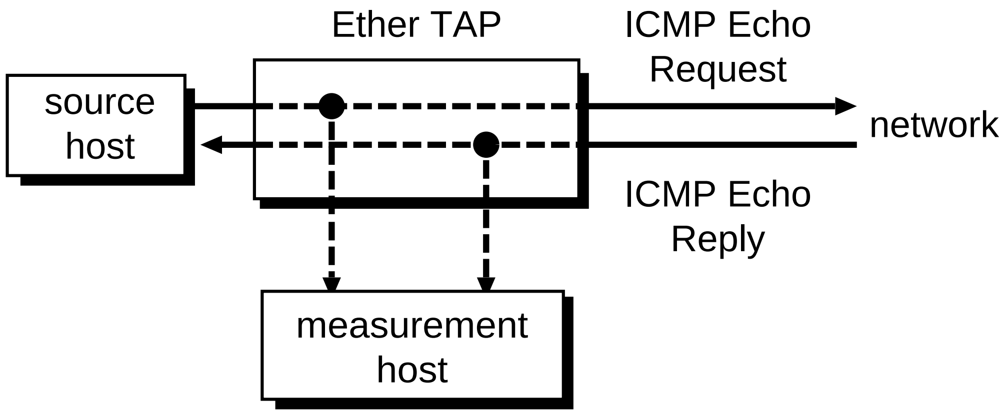


Figure 4: Measurement host for reliable data measurement.

contains the timestamp placed by the source host at its transmission time. This enables precise measurement of the round-trip time at the source host. Instead of measuring ICMP Echo Request/Reply packet sending/receiving time at the source, a *measurement host* is prepared (Fig. 4). It is for achieving reliable measurement even when the source host sends or receives packets at very high rate. As shown in Fig. 4, the *Ether TAP* copies all packets carried on the link, and sends copies to the measurement host; that is, all ICMP Echo Request/Reply packets sent from/to the source host are also delivered to the measurement host.

4.2 Network Environment

As the number of routers between source and destination hosts increases, the noise (i.e., effect of other traffic and measurement errors) contained in the output becomes large. Besides, the dominant part of the round-trip time is a queuing delay at the bottleneck router. It is therefore important to choose network configurations, in which the input and the output are collected, by taking account of the number of routers and the location of the bottleneck link. In this thesis, we measure packet sending/receiving times in three network configurations including LAN and WAN environments, and obtain the input $u(k)$ and the output $y(k)$. In the LAN environment, it is expected that the ARX model can accurately model the round-trip time dynamics since the network topology is rather simple and the measured data would suffer from little observation noise. On the contrary, in the WAN environment, it is expected that the model accuracy is degraded compared to that in the LAN environment since the network topology is complex. We use two network configurations for WAN environment. The difference of these WAN configurations is the location of the bottleneck link. In this thesis, the following three network configurations (i.e., **N1**, **N2**, and **N3**) are used for collecting input and output data.

- Network **N1** (LAN)

  The network **N1** is the LAN environment of a simple network configuration (Fig. 5). There exist two switches (SW1 and SW2) between source and destination hosts. All hosts and switches are connected to 100 Mbps LAN. The link between SW1 and SW2 also carries background traffic, as well as ICMP Echo Request/Replay packets exchanged between source and destination hosts. In this configuration, a bulk FTP transfer from a server (connected to SW1) to a client (connected to SW2) is performed during data collection.

- Network **N2** (WAN with the bottlenecked access link)

  The network **N2** is WAN environment of a complex network configuration, and the access link is the bottleneck between source and destination hosts (Fig. 6). The source host is connected to the Internet via 100 Mbps LAN, and the destination host is connected via 56 Kbps dial-up PPP link. At the time of measurement, the number of hops between source and destination hosts was 16, and the average round-trip time

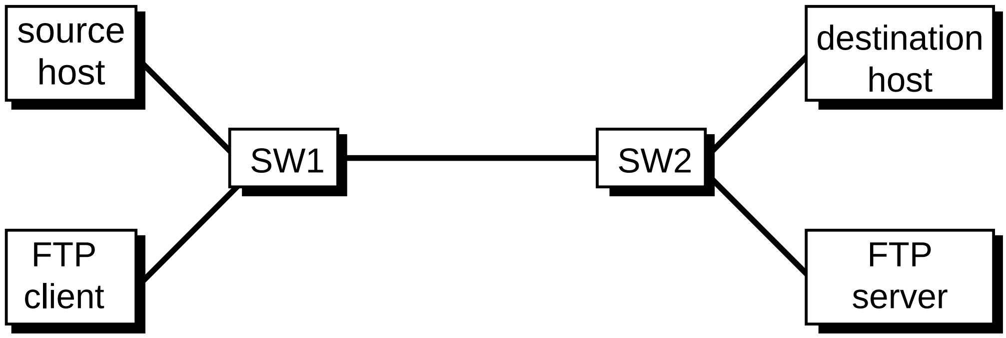


Figure 5: Network **N1** (LAN).

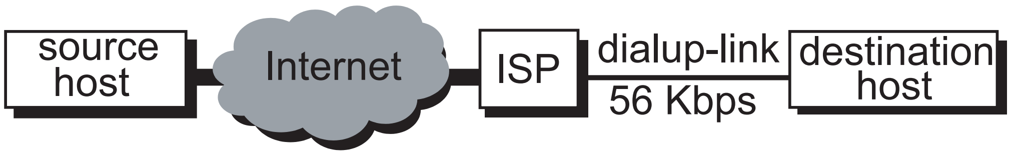


Figure 6: Network **N2** (WAN with the bottlenecked access link).

was 319.7 ms.

- Network **N3** (WAN with the non-bottlenecked access link)

  The network **N3** is WAN environment, and the access link is not the bottleneck between source and destination hosts (Fig. 7). The source host is connected to the Internet via 100 Mbps LAN. We choose `www.so-net.ne.jp` as a destination host. At the time of measurement, the number of hops between source and destination hosts was 16, and the average round-trip time was 36.89 ms.

In the above three network configurations, we measured the packet sending/receiving time at the measurement host. The source host sent 20,000 ICMP Echo Request packets, and the timestamp of each ICMP Echo Request/Replay packet was recorded by the measurement host. The data collection was done at midnight of October 18, 2001. As we have explained in Section 3, the input $u(k)$ and the output $y(k)$ for system identification is calcu-

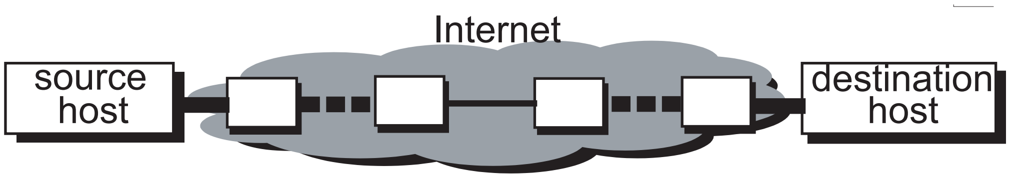


Figure 7: Network **N3**: (WAN with the non-bottlenecked access link).

lated from measured packet sending/receiving times. We empirically choose the sampling interval T in each network configuration; that is, T is chosen for each sampling period to contain about five samples. In this thesis, the packet inter-departure time from the source host is randomly changed, there might be a sampling period in which no packet is sent or received. If no packet is sent (or received) during kth sampling period, the input $u(k)$ (or the output $y(k)$) is not defined. In such a case, we use the minimum value of all past input (or output) data; i.e.,

$$
\begin{aligned}
u(k) &= \min_{0 \leq i < k} (u(k)) \\
y(k) &= \min_{0 \leq i < k} (y(k))
\end{aligned}
$$

5 Modeling from Measured Data

5.1 Choice of Model Orders and Number of Samples

The orders of the ARX model were fixed at $n_a = 5$ and $n_b = 5$ in all three network configurations. This is for comparing the model accuracy in each network configuration. The delay from the input to the output, n_d, is determined from the average round-trip time. This is because the packet sending rate at a specific time would have influence on the packet receiving process after the round-trip time. By letting N be the total number of round-trip time samples, n_d is determined as

$$n_d = \left\lfloor \frac{\sum_{k=1}^{N} y(k)}{N T} \right\rfloor$$

For evaluating the model accuracy, we use two validation methods: (1) a validation method using simulation, and (2) a validation method in frequency domain. The first method is to compare the simulated output from the ARX model, where zero noise is assumed (i.e., $e(k) = 0$), with the actual output [6]. The simulated output from the ARX model is defined in Eq. (3). The ARX model is thought to be accurate if the simulated output from the ARX model coincides to the actual output.

The second method is to compare the frequency response of the ARX model with the frequency response estimated by the spectral analysis [6]. The bode plot is used for visually comparing those frequency responses. The bode plot illustrates the gain and the phase of a dynamic system at different frequencies. When a linear stable dynamic system has a sinusoid input signal

$$u(t) \quad = \quad A\sin(\omega\, t),$$

the output from the system can be written as

$$y(k) \quad = \quad B\sin(\omega + \phi)$$

The gain and the phase of the system at the frequency ω are B/A and ϕ, respectively. The frequency response of the ARX model is easily obtained by deriving its corresponding transfer function. On the contrary, the spectral analysis directly estimates the frequency response from measured input and output data. The ARX model is thought to be accurate if frequency responses of the ARX model and the spectral analysis are identical.

From all input and output data obtained in Section 4, two datasets for determining parameters of the ARX model (i.e., model identification) and for validating the model accuracy (i.e., model validation) are extracted. The number of input and output data (i.e., the number of samples) used for system identification directly affects the model accuracy. We show the relation between the loss function and the number of input and output data, obtained from three network configurations. Figure 8 shows the relation between the *loss function* and the number of samples in the network configuration **N1**.

This figure indicates that the loss function becomes small as the number of input and output data increases. However, the loss function does not change However, the loss function will not change significantly if the number of samples exceeds about 150. Figure 9 shows the relation in network configuration **N2**. This figure indicates that the loss function is minimized when the number of samples is about 170. This figure also indicates that the

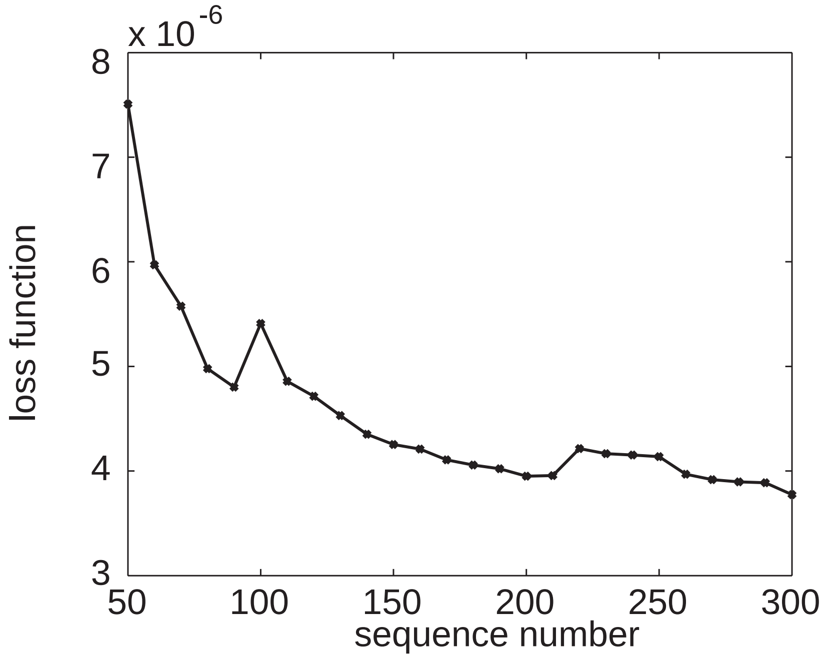


Figure 8: Relation between loss function and the number of input/output data in N1.

loss function increases as the number of samples exceeds about 170. Figure 10 shows the relation in network configuration **N3**. We can find that the loss function increases as the number of samples exceeds about 180.

We therefore use 150 input and output data for both model identification and model validation; i.e., we use input and output data from 2,001 to 2,150 for model identification, and from 2,201 to 2,350 for model validation.

5.2 Modeling Results and Discussions

- Network **N1** (LAN)

  Figure 11 shows (a) the input (the packet transmission rate from the source host), (b) the output (the average round-trip time). Figure 12 shows (a) comparison of the simulated output and the actual output, and (b) comparison of frequency responses

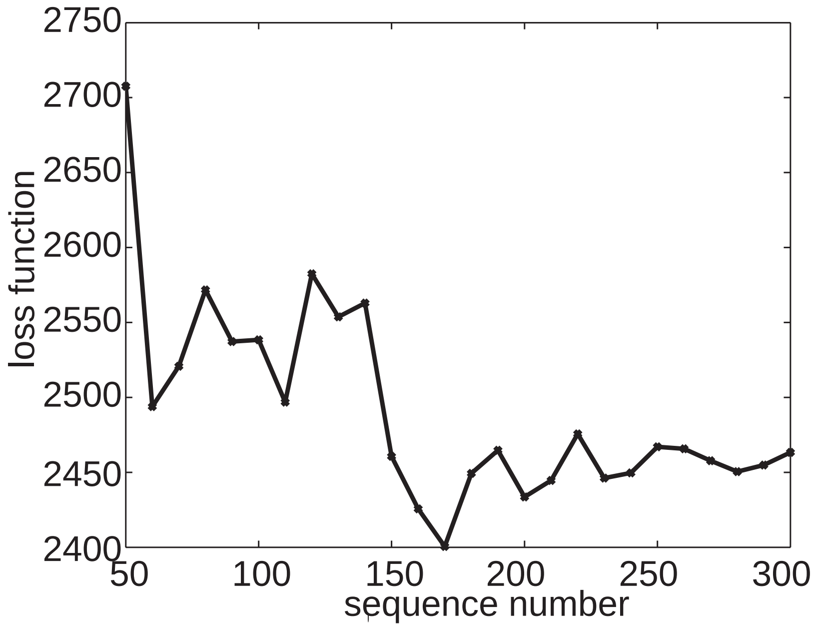


Figure 9: Relation between loss function and the number of input/output data in N2.

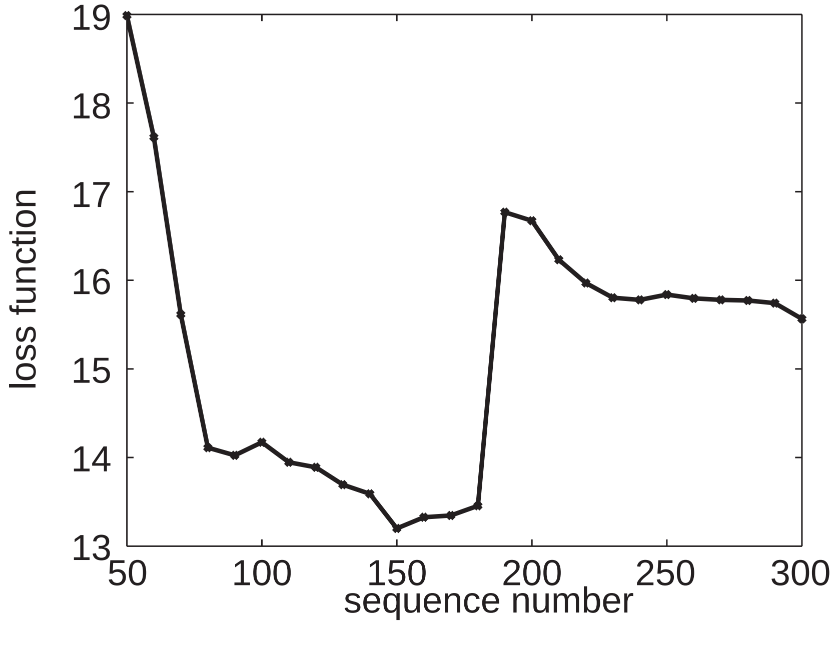


Figure 10: Relation between loss function and the number of input/output data in N3.

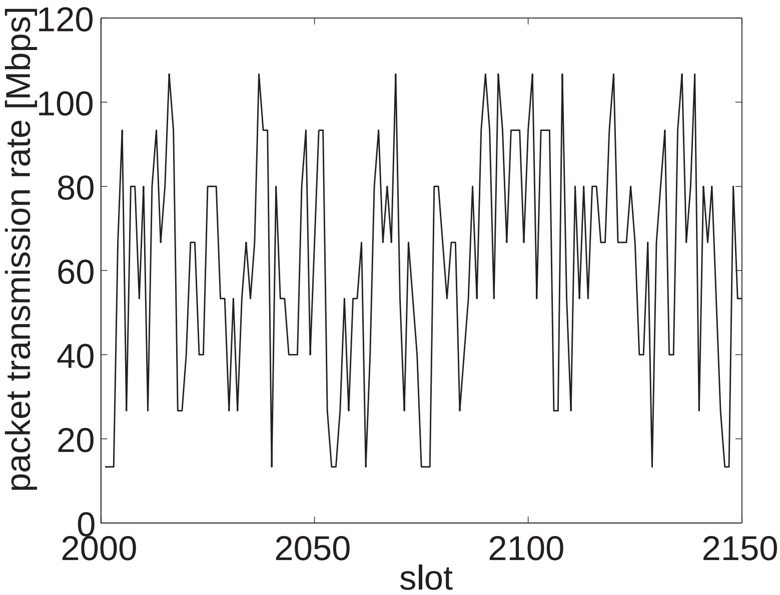
of the ARX model and the spectral analysis for the network **N1**. Note that (a) and (b) are input and output data not for model validation, but for model identification. In this case, the average packet transmission rate from the source host was 66.6 Mbps, and the average round-trip time was 0.42 ms. The average throughput of the FTP transfer was 5.2 Mbps. Also the sampling interval is $T = 0.9$ ms, and the delay of the ARX model is $n_d = 0$ according to Eq. (7).

Figure 12(a) shows a good agreement between the simulated output and the actual output. Figure 12(b) also shows a good agreement between frequency responses of the ARX model and the spectral analysis. Although frequency responses at a high frequency are different, such disagreement would be caused by inaccuracy of the spectral analysis, in particular, at a high frequency [6]. From these observations, we conclude that in the network **N1**, the round-trip time dynamics can be accurately modeled by the ARX model. Is is because that in the LAN environment, the packet transmission rate from the source host directly affects the packet waiting time at the bottleneck link, resulting in a strong correlation between the packet transmission rate and the round-trip time.

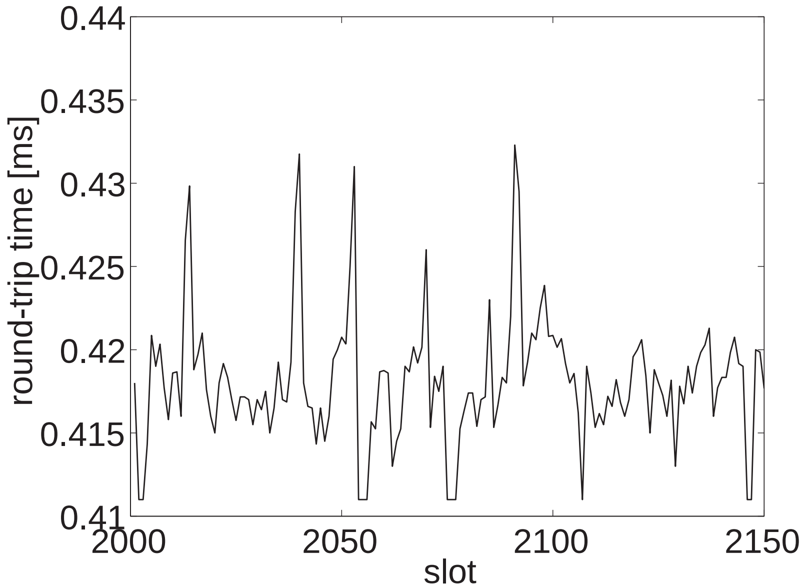
Recall that the average packet transmission rate in this experiment is rather high (i.e., 66.6 Mbps). Although results are not included here, the ARX model cannot capture the round-trip time dynamics when the average packet transmission rate was 20 Mbps. This phenomenon is possibly because of little packet waiting time at the bottleneck link.

- Network **N2** (WAN with the bottlenecked access link)

  Figure 13 shows input and output data for model identification, and Fig. 14 shows results of system identification for the network **N2**. In this experiment, the average

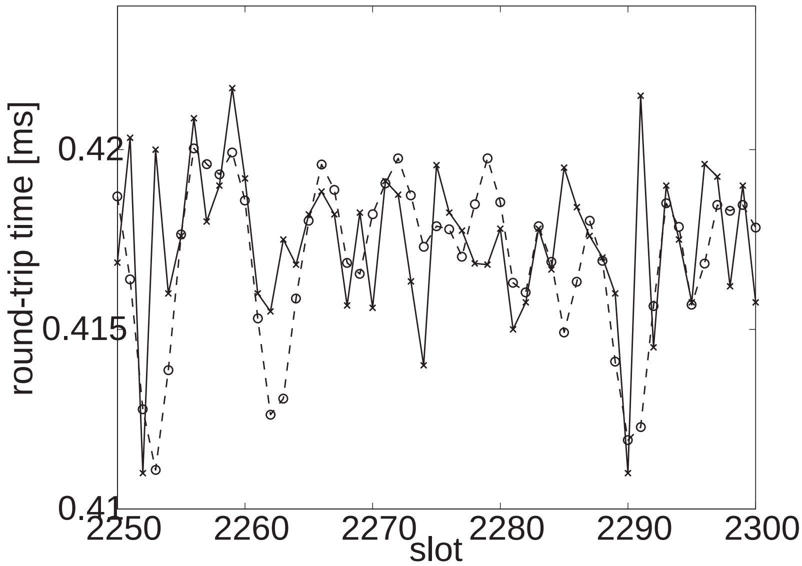


(a) Input data (packet transmission rate)

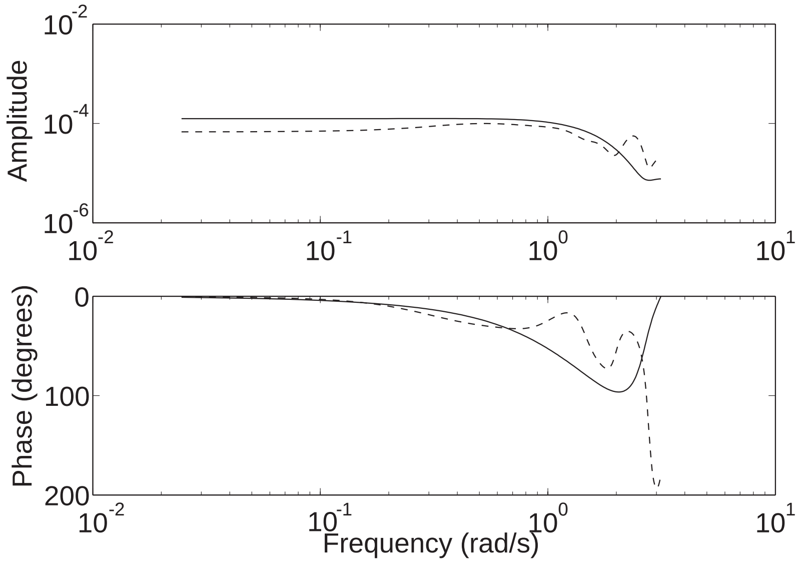


(b) Output data (average round-trip time)

Figure 11: Measurement Results of Network **N1**.



(a) Comparision with simulation (solid: measured output, dotted: simulated output)



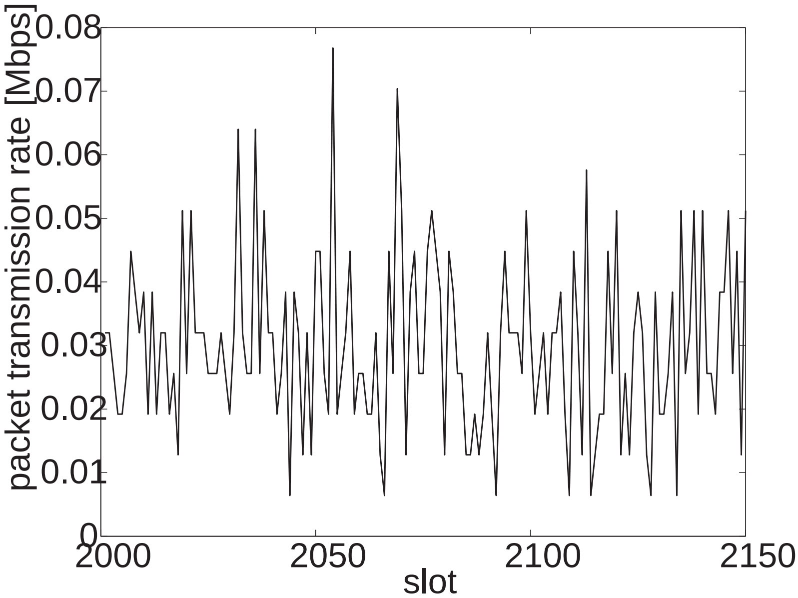
(b) Comparision in frequency domain (solid: ARX model, dotted: spectrum analysis)

Figure 12: Modeling Results of Network **N1**.

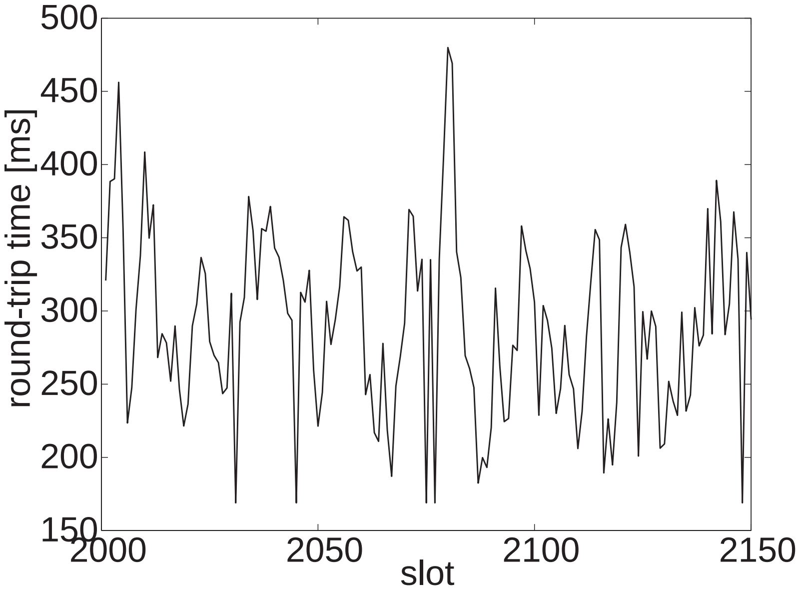
packet transmission rate from the source host was 31.8 Kbps, and the average round-trip time was 319.7 ms. The sampling interval T is 125 ms, and the delay of the ARX model n_d is 2 according to Eq. (7). The network **N2** is the WAN environment where there are 15 routers between source and destination hosts. Intuitively, it is expected that modeling the round-trip time dynamics is more difficult than in the network **N1**. However, Fig. 14(a) indicates that the simulated output and the actual output well coincides. In addition, frequency responses in Fig. 14(b) shows good agreement between the ARX model and the spectral analysis. Hence, the round-trip time dynamics can be accurately modeled by the ARX model in the network **N2**. This can be explained by the fact that the access link is the bottleneck between source and destination hosts. Namely, as the packet transmission rate from the source changes, the packet waiting time occurs at the bottleneck link. In the network **N2**, since the access link is the bottleneck, the round-trip time suffers little disturbance from other traffic. Therefore, there exists a strong correlation between the packet transmission rate and the round-trip time, resulting in an accurate ARX model. From these observations and discussions, we conclude that in WAN environment, the round-trip time dynamics can be accurately modeled when the bottleneck link is shared by a small number of users.

- Network **N3** (WAN with the non-bottlenecked access link)

  Figure 15 shows input and output data for model identification, and Fig. 16 shows results of system identification for the network **N3**, where the average packet transmission rate from the source host was 9.55 Mbps, and the average round-trip time was 36.89 ms. The sampling interval is $T = 6$ ms, and the delay of the ARX model is $n_d = 6$. Figure 16(a) shows that the simulated output and the actual output are

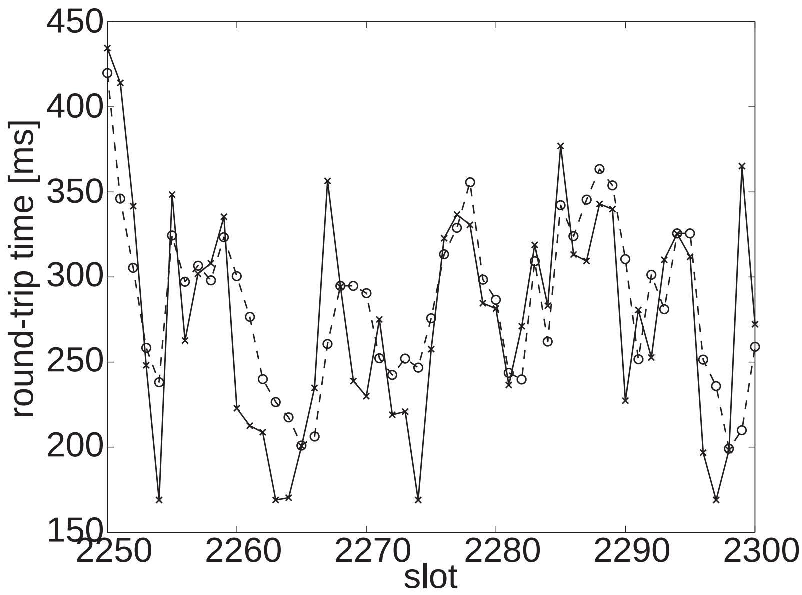


(a) Input data (packet transmission rate)

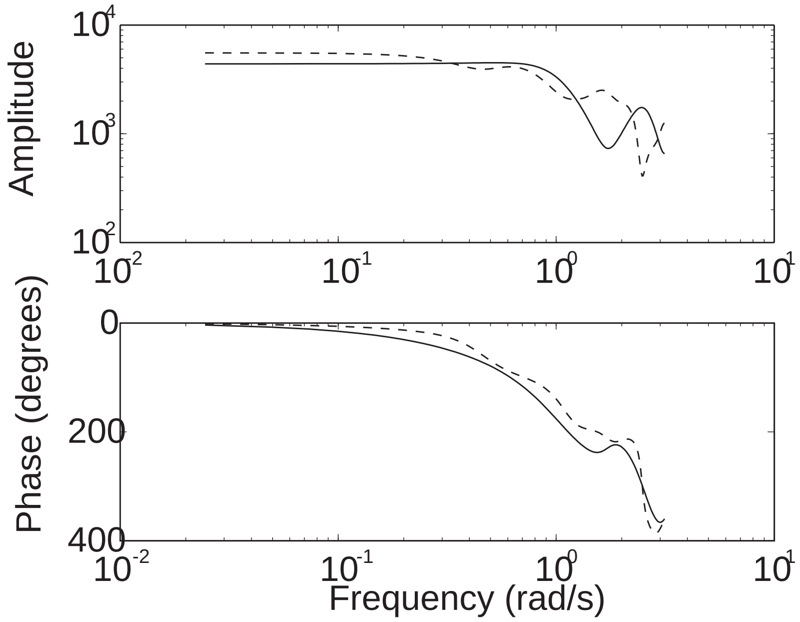


(b) Output data (average round-trip time)

Figure 13: Measurement Results of Network **N2**.

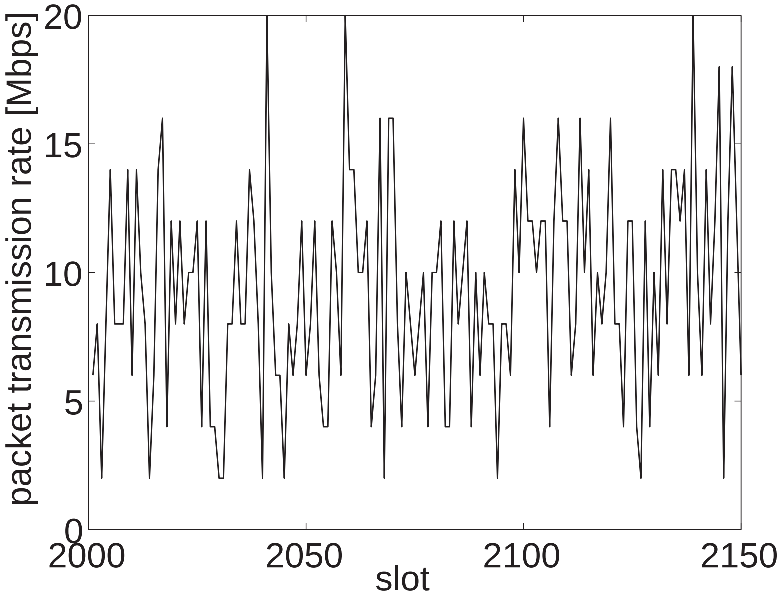


(a) Comparision with simulation (solid: measured output, dotted: simulated output)

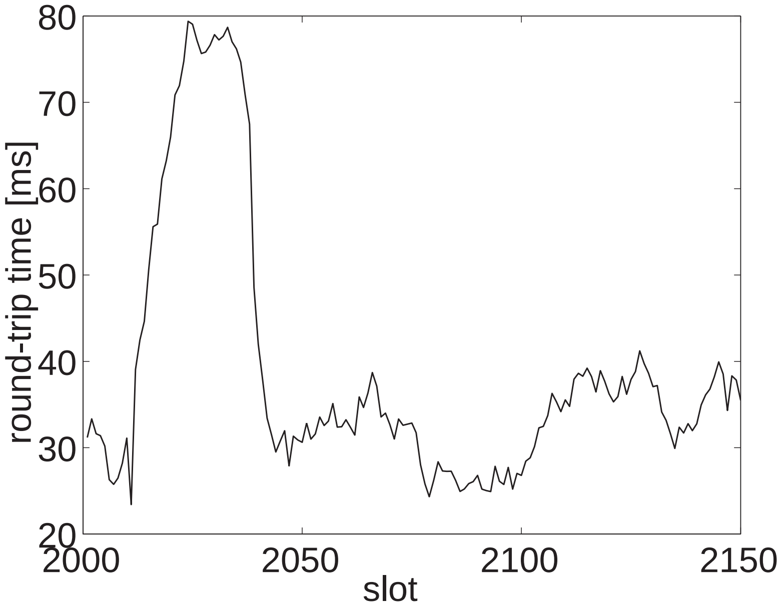


(b) Comparision in frequency domain (solid: ARX model, dotted: spectrum analysis)

Figure 14: Modeling Results of Network **N2**.

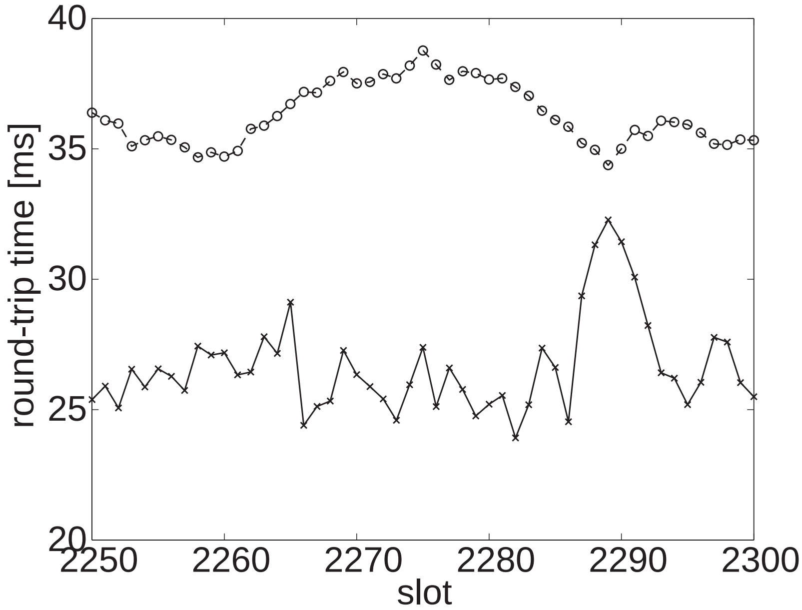


(a) Input data (packet transmission rate)

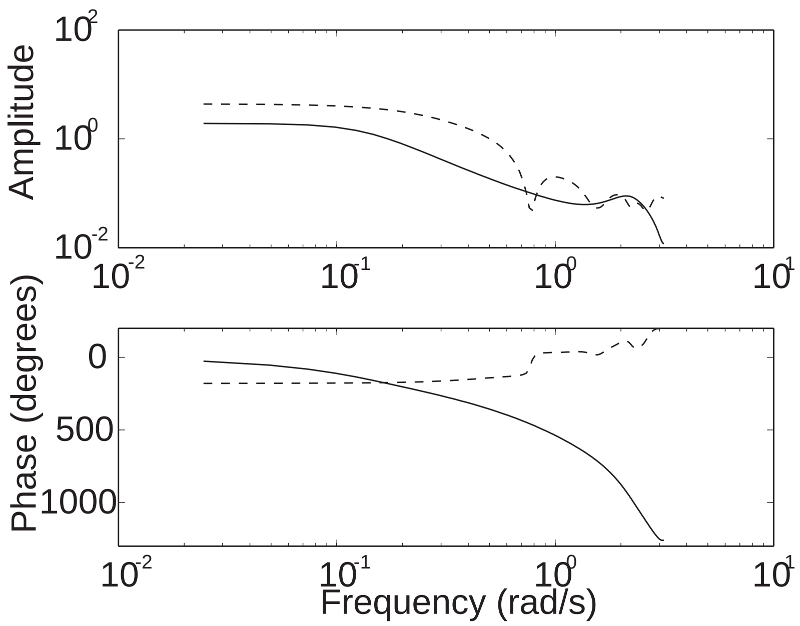


(b) Output data (average round-trip time)

Figure 15: Measurement Results of Network **N3**.



(a) Comparision with simulation (solid: measured output, dotted: simulated output)



(b) Comparision in frequency domain (solid: ARX model, dotted: spectrum analysis)

Figure 16: Modeling Results of Network **N3**.

completely different. Also, frequency responses in Fig. 16(c) disagree. Thus, in the network **N3**, the ARX model fails to capture the round-trip time dynamics. This inaccuracy would be caused by a large noise; that is, in the network **N3**, the network topology is rather complex, and the bottleneck link would be shared by many users. Hence, the packet transmission rate from the source host has little impact on the round-trip time. In other words, there is a very weak correlation between the packet transmission rate and the round-trip time, resulting in an inaccurate ARX model. Although we have conducted several experiments for different destination hosts, the ARX model cannot capture the round-trip time dynamics in any case. From these observations, we conclude that the ARX model is unable to model the round-trip time dynamics when the network configuration is complex and the packet transmission rate has little effect on the round-trip time.

Our findings indicate that the round-trip dynamics can be accurately modeled by the ARX model when the packet transmission rate from a source host has strong correlation to the measured round-trip time. This suggests that a delay-based congestion control between source and destination hosts is applicable when the round-trip time dyanamics can be modeled by our approach. In the following sections, we discuss several possible applications of our modeling approach, and design a simple rate control mechanism by applying the classical control theory.

6 Application Scenarios

I discuss several possible applications of our approach — modeling the round-trip time dynamics of the Internet using the ARX model. It is worthwhile to discuss how our approach can be applied to various problems. The first and straightforward application would be to use our approach to *understand* the round-trip time dynamics of the Internet. I can analyze the round-trip time dynamics through the ARX model. Because the ARX model is one of LTI (Linear Time Invariant) models, various analysis techniques for LTI models in time- and frequency-domain can be utilized. The second application would be to *predict* the future round-trip time from the ARX model obtained. As I have shown in Section 5, the round-trip time of a packet is considerably disturbed by background traffic. Hence, it is almost impossible to predict the far future round-trip time. However, the ARX model can predict the near future round-trip time. As noted in Section 1, the third and possibly most important application would be to *design* a delay-based congestion control mechanism. Once the ARX model capturing the round-trip time is obtained, it would be possible to apply the optimal control theory to design an efficient delay-based congestion control mechanism. Congestion control of the Internet is a difficult problem because of its complexity such as heterogeneity of various network elements and non-negligible propagation delays. In this section, I show the overview for designing a delay-based congestion control mechanism.

Figure 17 shows the block diagram of a rate-based congestion control mechanism which I design in this thesis. I regard the network seen by a specific source host as the control target (plant). I define a packet transmission rate as the output from the source host (the input to the network) and a round-trip time as the input to the source host (the output from

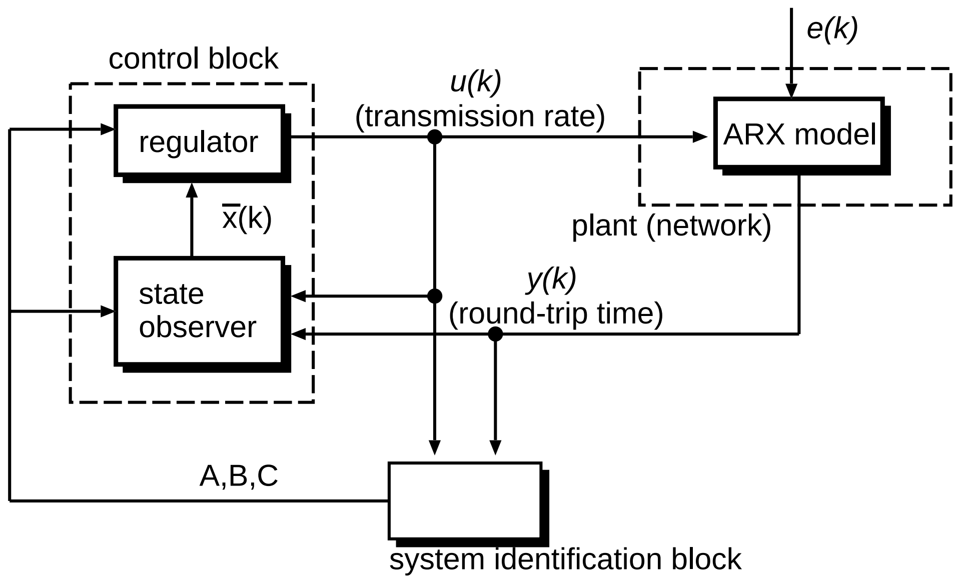


Figure 17: Block diagram of a rate-based congestion control mechanism.

the network). I assume that the destination host performs some procedure to measure the round-trip time for each packet at the source host. In this thesis, I design a delay based congestion control mechanism to stabilize the round-trip time for utilizing the network resources.

As depicted in Fig. 17, the source host is composed of two blocks (i.e., a system identification block and a control block). In a system identification block, the model capturing the round-trip dynamics can be obtained from the input and output data measured at the source host. A control block determines a packet transmission rate by applying the classical control theory to the model obtained in a system identification block.

For designing a rate-based congestion control mechanism, the control objective should be determined from various performance metrics about a congestion control mechanism. There are performance metrics such as,

- End-to-end throughput

- End-to-end packet loss probability
- End-to-end packet delay and its variance (jitter)

The rate-based congestion control mechanism I design controls a packet transmission rate using the variation of the round-trip time. Therefore, a control objective, I select, is to stabilize the round-trip time measured by the source host. If a control objective is properly selected, it is possible to reduce the packet delay and its variance. The queue length (the number of packets in a buffer) at the bottleneck router also can be decreased, so the bottleneck router can be utilized efficiently.

7 Conclusion

In this thesis, we have modeled the round-trip time dynamics of the Internet by the ARX (Auto-Regressive eXogenous) model using system identification. As input and output data for the ARX model, we have used the packet transmission rate from the source host and the average round-trip time measured by the source host. Using input and output data measured in working LAN and WAN environments, we have investigated how accurately the ARX model can capture the round-trip time dynamics. Through numerical examples, we have shown that in LAN environment, the round-trip time dynamics can be accurately modeled by the ARX model. We have also shown that in WAN environment, the round-trip time dynamics can be accurately modeled when the bottleneck link is shared by a small number of users.

As a future work, We are currently working to improve the model accuracy by using more complicated models such as ARMAX (Auto-Regressive Moving Average eXogenous) model. We are planning to design an efficient delay-based congestion control mechanism by utilizing the ARX model, which captures accurately the round-trip time dynamics.

Acknowledgments

I would like to express my sincere appreciation to Professor Masayuki Murata of Osaka University, who introduced me to the area of computer networks including the subjects in this thesis. All works of this thesis would not have been possible without the continuous support of Dr. Hiroyuki Ohsaki of Osaka University. It gives me great pleasure to acknowledge his effortless assistance. He has been constant sources of encouragement and advice through my studies and preparation of this manuscript. Thanks are also due to Professor Hideo Miyahara of Osaka University who gave me many insightful advices. I am also indebted to Dr. Naoki Wakamiya, Dr. Go Hasegawa and Mr. Shin'ichi Arakawa of Osaka University who gave me helpful comments and feedbacks. Finally, I thank many friends and colleagues in the Department of Informatics and Mathematical Science of Osaka University for their support.

References

[1] R. Jain, “A delay-based approach for congestion avoidance in interconnected heterogeneous computer networks,” *ACM Computer Communication Review*, vol. 19, pp. 56–71, Oct. 1989.

[2] L. S. Brakmo, S. W. O’Malley, and L. L. Peterson, “TCP Vegas: New techniques for congestion detection and avoidance,” in *Proceedings of ACM SIGCOMM ’94*, pp. 24–35, Oct. 1994.

[3] J.-C. Bolot, “Characterizing end-to-end packet delay and loss in the Internet,” *Journal of High-Speed Networks*, vol. 2, pp. 305–323, Dec. 1993.

[4] S. B. Moon, J. Kurose, P. Skelly, and D. Towsley, “Correlation of packet delay and loss in the Internet,” tech. rep., Department of Computer Science, University of Massachusetts, USA, Jan. 1998.

[5] V. Paxson, “End-to-end Internet packet dynamics,” in *Proceedings of ACM SIGCOMM ’97*, pp. 139–152, Sept. 1997.

[6] L. Ljung, *System identification — theory for the user*. Englewood Cliffs, N.J.: Prentice Hall, 1987.

[7] M. Allman and V. Paxon, “On estimating end-to-end network path properties,” in *Proceedings of ACM SIGCOMM ’99*, pp. 263–274, Aug. 1999.

[8] A. Adams, T. Bu, R. Caceres, N. Duffield, T. Friedman, J. Horowitz, F. L. Presti, S. B. Moon, V. Paxson, and D. Towsley, “The use of end-to-end multicast measurements for characterizing internal network behavior,” *IEEE Communications*, pp. 152–159, May 2000.

[9] V. Paxson, “End-to-end routing behavior in the Internet,” in *Proceedings of ACM SIGCOMM ’96*, pp. 25–38, May 1996.

[10] R. Caceres, P. B. Danzig, S. Jamin, and D. J.Mitzel, “Characteristics of wide-area TCP/IP conversations,” in *Proceedings of ACM SIGCOMM ’91*, pp. 101–112, Sept. 1991.

[11] H. Che and S.-Q. Li, “Fast algorithms for measurement-based traffic modeling,” *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 612–625, June 1998.

[12] L. A. Kulkarni and S.-Q. Li, “Measurement-based traffic modeling: capturing important statistics,” *Journal of Stochastic Model*, vol. 14, no. 5, 1998.

[13] S.-Q. Li, S. Park, and D. Arifler, “SMAQ: A measurement-based tool for traffic modeling and queueing analysis, Part I - design methodologies and software architecture,” *IEEE Communications Magazine*, vol. 36, pp. 56–65, Aug. 1998.

[14] S.-Q. Li, S. Park, and D. Arifler, “SMAQ: A measurement-based tool for traffic modeling and queueing analysis, Part II - network applications,” *IEEE Communications Magazine*, vol. 36, pp. 66–77, Aug. 1998.

[15] O. Ait-Hellal, E. Altman, and T. Basar, “A robust identification algorithm for traffic models in telecommunications,” in *Proceedings of IEEE CDC ’99*, Dec. 1998.

[16] A. Sang and S.-Q. Li, “A predictability analysis of network traffic,” in *Proceedings of IEEE INFOCOM 2000*, pp. 342–351, Mar. 2000.

[17] S. Hares, “Essential tools for the OSI Internet,” *Request for Comments (RFC) 1574*, Feb. 1994.

[18] F. Baker, "Requirements for IP version 4 routers," *Request for Comments (RFC) 1812*, June 1995.

[19] J. Postel, "Internet control message protocol," *Request for Comments (RFC) 792*, Sept. 1981.

[20] S. Savage, "Sting: A TCP-based network measurement tool," in *Proceedings of USENIX Symposium on Internet Technologies and Systems*, pp. 71–79, Oct. 1999.